BAB II
TINJAUAN PUSTAKA

2.1 Studi Literatur

Berikut adalah beberapa penelitian terdahulu yang terkait dengan tema komparasi algoritma klasifikasi, secara garis besar tinjauan pustaka dalam tesis ini meliputi:

a. Perbandingan Algoritma Decision Trees dan Naïve Bayes untuk Analisis Faktor Penyebab Kebangkrutan Perusahaan (Bima Putra Amiga, 2018), Penelitian ini menggunakan algoritma Decision Trees dan Naïve Bayes untuk dibandingkan nilai akurasinya, yang akan berguna untuk menganalisis faktor – faktor yang dapat menyebabkan bangkrutnya sebuah perusahaan. Adapun kesimpulan yang diperoleh dari hasil analisis kedua algoritma ini adalah algoritma Decision Tree memiliki nilai akurasi yang lebih tinggi dari algoritma Naive Bayes dalam metode pembagian data secara Cross Validating sebesar 93% dan secara Bootstrapping dengan nilai rata-rata akurasi sebesar 92,5%. Sedangkan untuk naïve bayes hanya pada rata-rata akurasi sebesar 52, 8 %.

b. Komparasi Algoritma Naive Bayes dan Decision Tree Untuk Memprediksi Lama Studi Mahasiswa (Indera Cahyo Wibowo, 2019), Penelitian ini bertujuan memprediksi lama studi mahasiswa lulus tepat waktu atau terlambat lulus. Data yang digunakan yaitu: gender, status mahasiswa, nilai, dan beasiswa dari semester awal sampai semester akhir di tahun ajaran 2018 – 2019. Penelitian ini menggunakan 2 metode, untuk metode yang pertama adalah Naïve Bayes dan yang kedua adalah Decision Tree. Gunakan semua data yang telah diperoleh dan hitung dengan kedua metode sampai mendapatkan hasil akhir dan akurasinya lalu komparasikan keduanya. Hasil pengkomparasian dari data dan menggunakan dua metode perhitungan keakurasiannya yaitu 30% untuk Naïve bayes, dan 55% untuk metode Decision Tree. Dari komparasi menggunakan dua metode dapat diambil kesimpulan bahwa metode Decision Tree memiliki persentase keakuratan

yang lebih tinggi dibandingkan Naïve Bayes sehingga dapat dikatakan metode Decision Tree lebih akurat dan lebih detail daripada Naïve Bayes.

c. Perbandingan kinerja metode C4.5 dan Naive Bayes dalam klasifikasi artikel jurnal PGSD berdasarkan mata pelajaran (Utomo Pujianto, 2019). penelitian ini untuk mengklasifikasikan minat mahasiswa PGSD terhadap tema mata pelajaran menurut SK-KMP menggunakan metode Naive Bayes dan Decision tree J48. Hasil penelitian tersebut dapat dijadikan sebagai referensi untuk pengambilan tema pada mata pelajaran di tahun mendatang untuk lebih bervariasi, tidak hanya membahas tentang salah satu mata pelajaran tersebut. Kinerja dari kedua metode tersebut akan dibandingkan, sehingga dapat diketahui kinerja metode mana yang lebih baik dalam melakukan klasifikasi dokumen. Pengujian performa algoritma klasifikasi yang digunakan adalah teknik K-fold Cross Validation. Berdasarkan pengujian performa penerapan algoritma Naïve Bayes dan Decision Tree J48 menggunakan teknik K-Fold Cross Validation terhadap 200 judul dan abstrak artikel jurnal, didapatkan algoritma Naive Bayes, tingkat akurasi sebesar 84%. Sementara itu, untuk hasil yang diperoleh dengan algoritma Decision Tree J48, tingkat akurasi sebesar 86%.

d. Perbandingan Algoritma Klasifikasi Data Mining Model C4.5 Dan Naive Bayes Untuk Prediksi Penyakit Diabetes (Fatmawati, 2016), Penelitian ini bertujuan membuat klasifikasi dan menerapkan klasifikasi data mining. Hasil klasifikasi data di evaluasi dengan menggunakan Confusion Matrix dan kurva ROC untuk mengetahui tingkat hasil akurasi menggunakan algoritma Decision Tree yaitu sebesar 73.30% dan nilai AUC dari kurva ROC adalah 0.733 sedangkan algoritma Naive Bayes sebesar 75.13% nilai AUC dari kurva ROC 0.810 sehingga dapat dikatakan bahwa algoritma Naive Bayes memiliki hasil prediksi yang baik dalam memprediksi penyakit diabetes seorang pasien.

e. Metode Pembelajaran Mesin Untuk Deteksi Dan Klasifikasi Malware (Kateryna Chumachenko, 2017), membahas tentang bagaimana mengklasifikasikan malware dengan metode pembelajaran mesin. Pada

penelitian ini hasil klasifikasi dari K-Nearest Neighbor, Support Vector Machine, J48 Decision Tree, Naïve Bayes dan Random Forest akan dibandingkan hasilnya. Pengujian dilakukan dengan software Weka dengan menggunakan dataset yang terdiri dari 1.156 file malware. Diperoleh hasil sebagai berikut :

1) Algoritma KNN menghasilkan akurasi yang baik sebesar 87% untuk klasifikasi kelas jamak/multi-class dan 94.6% untuk klasifikasi dua kelas/two-class.
2) Algoritma SVM menghasilkan akurasi yang baik sebesar 87,6% untuk klasifikasi kelas jamak/multi-class dan 94.6% untuk klasifikasi dua kelas/two-class.
3) Algoritma J48 Decision Tree menghasilkan akurasi yang baik sebesar 93,3% untuk klasifikasi kelas jamak/multi-class dan 94.6% untuk klasifikasi dua kelas/two-class.
4) Algoritma Naïve Bayes menghasilkan akurasi yang baik sebesar 72,23% untuk klasifikasi kelas jamak/multi-class dan 55% untuk klasifikasi dua kelas/two-class.
5) Algoritma Random Forest menghasilkan akurasi yang baik sebesar 95,69% untuk klasifikasi kelas jamak/multi-class dan 96,8% untuk klasifikasi dua kelas/two-class.

f. Pendeteksian Malware pada Lingkungan Aplikasi Web dengan Kategorisasi Dokumen (Fransiskus Gusti Ngurah Dwika Setiawan, 2017), Proses kategorisasi dokumen meliputi praproses dan tokenisasi kode sumber, pembuatan model classifier Multinomial Naive Bayes dan Decision Tree, dan klasifikasi dokumen menggunakan classifier yang telah dibuat. Uji coba yang dilakukan terhadap 718 file kode sumber PHP. Dari hasil uji coba dengan 218 file malicious dan 500 file non-malicious didapat bahwa Multinomial Naive Bayes dapat mengklasifikasi dengan tingkat precision sebesar 83% dan recall sebesar 83%. Sedangkan model Decision Tree dapat mengklasifikasi dengan tingkat precision sebesar 72% dan recall hingga 97%. Model classifier Multinomial Naive Bayes cenderung lebih presisi

daripada Decision Tree, namun Decision Tree dapat lebih banyak mendeteksi file-file malicious.

g. Penerapan Naïve Bayes Pada Pendeteksian Malware Dengan Diskritisasi Variabel (Inda Anggraini, 2019), membahas tentang analisa terhadap serangan malware dengan menggunakan Algoritme Naïve Bayes Clasiffier dengan diskritisasi variabel Min-Max diskritisasi 3- interval dan 5-interval untuk atribut kontinu. Pengujian menggunkan software Rapidminer, diperoleh hasil percobaan menunjukan bahwa penerapan Naïve Bayes pada klasifikasi data yang belum melalui tahap pendiskritan menghasilkan tingkat akurasi sebesar 69.72 % dengan prediksi malware 63.53 % sedangkan pada data yang telah melewati tahap diskritisasi mampu memberikan akurasi hingga 79.97 % dengan prediksi malware 81.29 %. Penggunaan metode Naïve Bayes dalam penelitian ini memiliki kemampuan deteksi yang meningkat dibandingkan dengan proses klasifikasi tanpa menggunakan proses binning (diskritisasi). Proses pendiskritan dapat menjadikan Algoritme Naïve Bayes menjadi lebih akurat di dalam mendeteksi malware

h. Klasifikasi Malware Menggunakan Teknik Reverse Engineering dan Data Mining (Ravindar Reddy Ravula, 2011), penelitian ini membahas tentang klasifikasi malware dengan algoritma Decision Tree dan Naïve bayes dengan menerapkan teknik diskritisasi. Dataset yang digunakan peneliti, dikumpulkan dari internet dari situs www.mcaffe.com dan www.messagelabs.com. Dataset yang digunakan untuk klasifikasi data mining adalah hasil preprosessing data mentah yang diperoleh dari proses sebelumnya dengan menggunakan teknik Reverse/Reverse Engineering. Data Terdiri dari 2 (dua) set :

    1) **Dataset DRF (Dataset with Reverse Feauture)** yang terdiri dari 15 Fitur dan 1.103 Instance (582 Malware dan 521 benign software)
    2) **Dataset DAF (Dataset with API Call Features)** yang terdiri dari 141 Fitur dan 1.103 Instance (582 Malware dan 521 benign software)

Selanjutnya pada kedua dataset tersebut tidak dilakukan normalisasi nilai (penskalaan pada nilai atributnya), tetapi hanya mengurangi fitur/atribut

yang tidak diperlukan lagi(atribut reduction) pada dataset DRF yang semula 15 atribut menjadi 12 atribut saja, kemudian dataset DRF tersebut di proses Diskritisasi dengan teknik Entropy-Based Discretization dengan Tool Weka sehingga menghasilkan dataset DRF (Dataset with Discretized Feature), sehingga jumlah dataset menjadi 3 (tiga) yakni DRF, DAF dan DDF yang diuji dengan algoritma Decision Tree J48 dan Naïve Bayes, dimana masing-masing dataset tersebut dibagi menjadi 80% Data Training dan 20% Data Testing, dengan hasil sebagai berikut :

Tabel 2.1 Hasil Pengujian Penelitian Sebelumnya

No	Dataset	TP	TN	FP	FN	ROC Area	Oveall Accuracy	Algoritma	Ket
1	DRF	95	85	25	16	0.889	81,448%	Decision Tree J48	
2	DRF	88	89	21	23	0.843	80.090%	Naïve Bayes	
3	DAF	89	108	6	18	0.917	89.140%	Decision Tree J48	
4	DAF	75	108	6	32	0.921	85,067%	Naïve Bayes	
5	DDF	94	86	24	17	0.815	81.448%	Decision Tree J48	Dataset DRF didiskritisasi
6	DDF	96	92	18	15	0.912	85,067	Naïve Bayes	Dataset DRF didiskritisasi

Hasil pengujian pada tabel di atas terlihat dataset DRF akurasi Decision Tree berada di akurasi 81,448% lebih baik jika dibandingkan dengan naïve bayes yang berada pada angka 80,090% sebaliknya jika menggunakan Dataset DDF(DRF didiskritisasi), menghasilkan akurasi naïve bayes 85,067% yang lebih baik dari decision tree hanya 81,448%. Pada penelitian ini, menghasilkan akurasi naïve bayes lebih baik dari decision tree jika menerapkan teknik diskritisasi tetapi sebaliknya jika pengujian tanpa teknik diskritisasi diperoleh hasil akurasi decision tree lebih baik dari naïve bayes.

Dari beberapa penelitian terkait yang telah disebutkan di atas, maka didapatkan *state of the art* sebagai berikut :

Tabel 2.2 State Of The Art

No	Peneliti	Algoritma	Data	Software
1.	Bima Putra Amiga (2018)	▪ Decision Tree Kriteria Information Gain ▪ Multinomial Naïve Bayes (Data Kategorikal) ▪ Gaussian Naive Bayes (Data Numerik)	▪ Numerik ▪ Kategorik	Weka Tools
2.	Indera Cahyo Wibowo (2019)	▪ Decision Tree Kriteria Information Gain ▪ Multinomial Naïve bayes	Kategorik	
3.	Utomo Pujianto (2019)	▪ Decision Tree Kriteria Information Gain ▪ Multinomial Naïve bayes	Kategorik	Weka Tools
4.	Fatmawati (2016)	▪ Decision Tree Kriteria	Kategorik	Rapidminer Tools

No	Peneliti	Algoritma	Data	Software
		Information Gain ▪ Multinominal Naïve bayes		
5.	Kateryna Chumachenko (2017)	▪ Decision Tree J48 ▪ Gaussian Naive Bayes	Numerik	Weka Tools
6.	Fransiskus Gusti Ngurah Dwika Setiawan (2017)	▪ Multinomial Naive Bayes ▪ Decision Tree CART	Kategorik	Web PHP
7.	Inda Anggraini (2019)	Gaussian Naive Bayes	Numerik (Diskritisasi dengan binning)	Rapidminer
8.	Ravindar Reddy Ravula (2011)	▪ Decision Tree J48 ▪ Multinomial Naive Bayes	Kategorik (Diskritisasi berbasis entropy)	Weka Toos

Selanjutnya peneliti pada penelitian ini akan membahas tentang perbandingan algoritma decision tree dan naïve bayes pada pendeteksian malware dengan diskritisasi variabel, pada penelitian ini akan menggunakan jenis algoritma decision tree C4.5 dengan menggunakan kriteria gini index sedangkan untuk naïve bayes model gaussian yang akan digunakan. Adapun dataset yang akan digunakan peneliti berasal dari dataset malware bersumber dari website kaggle milik saravana tahun 2018, https://www.kaggle.com/saravana/malware-detection yang terdiri dari 35 atribut dan 100.000 instance yang berisi data numerik(kontinyu), kemudian data

tersebut dilakukan pre-processing dengan normalisasi dengan fungsi Z-score lalu dilakukan reduksi atribut dengan metode analisa korelasi dengan menghilangkan atribut-atribut yang tidak kontributif, setelah itu di proses dengan teknik diskritisasi dengan metode binning menggunakan Tool Rapidminer sehingga terbentuk dataset yang telah diskritisasi yang selanjutnya akan diuji dengan algoritma Decesion Tree C4.5 Gini Index dan Gaussian Naïve Bayes dengan distribusi dataset masing-masing 80% Data Training dan 20 % Data Testing, pengujian ini akan dilakukan dengan Tool Rapidminer 9.3 agar diperoleh algoritma mana yang memiliki tingkat akurasi terbaik.

2.2 Data Mining

Data mining adalah teknologi yang mengombinasikan metode analisis tradisional dan algoritma yang canggih agar proses data besar lebih cepat diproses. Data mining biasa disebut dengan sebutan yang sering digunakan untuk mencari pengetahuan yang tersembunyi didalam database. Data mining menggunakan teknik statistika, matematika, kecerdasan buatan, dan machine learining untuk mengekstraksi dan menganalisa informasi yang terdapat dalam database besar. (Turban et al, 2005).

Analisis yang dilakukan data mining lebih maksimum dibandingkan sistem pendukung keputusan tradisional yang banyak digunakan.Data mining mengatasi masalah-masalah bisnis dengan cara tradisional yang menggunakan banyak waktu da biaya yang tinggi. Data mining menjelajahi basis data untuk mengetahui pola yang tersembunyi, serta mencari informasi agar dapat memprediksi yang bisa saj dilupakan oleh pembisnis karena kemungkinan besar mereka tidak menduganya.

Pada perkembangan teknologi saat ini, proses pengumpulan data serta penyimpanannya telah mudah dijalankan walaupun data tersebut berukuran besa sehingga data mining melakukan proses pencarian secara mudah dan otomati mencari informasi yang berguna dalam penyimpanan data yang mempunyai ukura yang besar. Istilah ini biasa disebut denganKnowledge Discovery in Database(KDD) yang digunakan secara bergantian untuk memberikan penjelasan

tentang prose pencarian informasi yang tersembunyi dalam suatu basis data yang besar. Sebenarny konsep ini berkaitan satu sama lain walaupun konsepnya berbeda.

2.3 Teknik Data Mining

Data Mining merupakan proses untuk mencari nilai tambah dari beberpa data yang tidak bisa dilakukan secara manual, *Data Mining* menganalisa secara otomatis dari data yang berukuran sangat besar dengan fungsi untuk menemukan pola yang sangat penting terkadang tidak diketahui keberadaannya, dan *Data Mining* dapat memilih informasi yang bermanfaat dari database yang tersembunyi yang sebelumnya tidak dikenali. Proses ini mendekati teknis yang berbeda seperti *Clustering, Data Summarization, Learning Classification, Rules.* Tidak semua proses disebut *Data Mining* dalam mencari informasi, misalnya pencarian rekaman individu menggunakan *database management system* atau pencarian *web* yang menggunakan *query* kesemua *search engine* yang berkaitan dengan information *retrieval* dan *data mining* digunakan untuk meningkatkan kemampuannya.

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Perlu diingat bahwa kata *mining* sendiri berarti usaha untuk mendapatkan sedikit data berharga dari sejumlah besar data dasar. Karena itu *data mining* sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan *(artificial intelligent), machine learning,* statistik dan basisdata. Beberapa teknik yang sering disebut-sebut dalam literatur *data mining* antara lain yaitu *association rule mining,clustering, klasifikasi, neural network, genetic algorithm* dan lain-lain.

Klasifikasi biasanya berhubungan dengan peramalan kategori kelas dan menggolongkan data atau membangun sebuah model yang berdasarkan dengan pelatihan data untuk menetapkan dan nilai-nilai kelas dalam sebuah golongan atribut dan menggunakan golongan data baru. klasifikasi sering digunakan dalam bidang persetujuan kredit, target marketing, diagnose medis, dan analisa keefektifan sebuah keputusan. Langkah klasifikasi dengan menguraikan sebuah himpunan kelas yang telah ditentukan dan menggunakan model yang berfungsi untuk mengklasifikasi tuple data yang label kelasnya belum diketahui. Model-

model tersebut disajikan sebagai kaidah klasifikasi, pohon keputusan, atau rumus matematis. Macam-macam klasifikasi yang sering digunakan adalah *Decision Tree, Bayesian Network, Adaptive Bayesian Network, Naïve Bayes, Random Forest, Random Tree* dan lain sebagainya.

2.4 Tahapan Data Mining

Salah satu tuntut an dari*data mining* ketika diterapkan pada data berskala besar adalah diperlukan metodologi sistematis tidak hanya ketika melakukan analisa saja tetap juga ketika mempersiapkan data dan juga melakukan interpretasi dari hasilnya sehingga dapat menjadi aksi ataupun keputusan yang bermanfaat. *Data mining* seharusnya dipahami sebagai suatu proses, yang memiliki tahapan-tahapan tertentu dan juga ada umpan balik dari setiap tahapan ke tahapan sebelumnya. Pada umumnya proses *data mining* berjalan interaktif karena tidak jarang hasil data mining pad awalnya tidak sesuai dengan harapan analisnya sehingga perlu dilakukan desain ulan prosesnya.

Sesuai yang tercantum dalam buku "Advances in Knowledge Discovery dan Data mining" terdapat definisi sebagai berikut:*Knowledge discovery (data mining) in databases* (KDD) adalah keseluruhan proses *non-trivial* untuk mencari dan mengidentifikasi pola *(pattern)* dalam data, dimana pola yang ditemukan bersifat sah *(valid),* baru *(novel),* dapat bermanfaat *(potentially usefull),* dapat dimengerti *(ultimately understandable).* (Vapnik, 1998). Istilah *data mining* dan *Knowledge Discovery In Databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbedaakan tetap berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining.* Proses KDD secara garis besar dapat dijelaskan sebagai berikut:

a. Data Selection

   Pemilihan (seleksi) data dan sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi

yang akan digunakan untuk proses *data mining,* disimpan dalam suatu berkas, terpisah dari basis data operasional.

b. *Pre-processing/Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment,* yaitu pros es "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

c. *Transformation Coding*

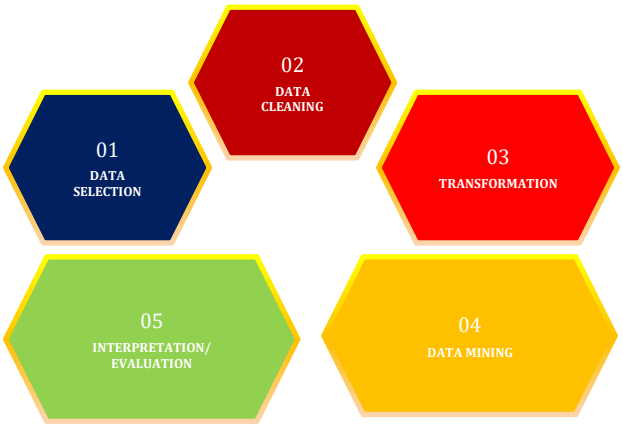
Proses transformasipada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

d. Data mining

Proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi.

e. Interpretation / evalution

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.



Gambar 2.1 Tahapan Data Mining

2.5 Algoritma Klasifikasi Data Mining

Algoritma Klasifikasi Data Mining Klasifikasi adalah proses penemuan model (atau fungsi) yang membedakan dan menggambarkan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui. Klasifikasi data terdiri dari 2 langkah proses. Pertama adalah learning (fase training), dimana algoritma klasifikasi dibuat untuk menganalisa data training lalu direpresentasikan dalam bentuk rule klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari rule klasifikasi. Proses klasifikasi didasarkan pada empat komponen:

a. Kelas

   Variabel dependen yang berupa kategorikal yang merepresentasikan ”label“ yang terdapat pada objek. Contohnya: risiko penyakit jantung, risiko kredit, customer loyalty, jenis gempa.

b. Prediktor

   Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.

c. Training dataset Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan predictor.
d. Testing dataset Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

Berikut ini adalah algoritma klasifikasi yang banyak digunakan secara luas:

a. Decision/classification trees

   Decision tree digunakan untuk memprediksi keanggotaan suatu objek ke dalam kategori (kelas) yang berbeda, berdasarkan variabel prediktor. Algoritma decision tree yang dikenal luas antara lain Hunt, CART (C&RT), ID3, C4.5&C5.0, SLIQ, SPRINT, QUEST, DTREG, THAID, CHAID.

b. Bayesian classifiers/ Naïve Bayes classifiers

   Klasifikasi Bayes adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. Klasifikasi Bayes juga dikenal dengan Naïve Bayes, idiot's Bayes, simple Bayes, dan independence Bayes. Klasifikasi Bayes berdasarkan pada teorema Bayes, diambil dari nama seorang ahli matematika yang juga menteri Prebysterian Inggris, Thomas Bayes (1702-1761).

c. Neural networks

   Neural network adalah (Han, 2006) satu set unit input/output yang terhubung dimana tiap relasinya memiliki bobot. Neural Network dimaksudkan untuk mensimulasikan perilaku sistem biologi susunan syaraf.

d. K-Nearest Neighbor

   K-Nearest Neighbor (kNN) adalah klasifikasi yang menyimpan semua data training dan melakukan klasifikasi dengan cara membandingkan antara atribut data baru yang paling cocok dengan atribut record yang terdapat pada data training. kNN adalah kasus khusus dalam instancebased learning. Ini termasuk case-based reasoning, yang menangani data simbol. kNN juga merupakan contoh teknik lazy learning, yaitu teknik yang menunggu sampai pertanyaan (query) datang agar sama dengan data training.

e. Support vector machines (SVM)

Support vector machines (SVM) dibuat oleh Vapnik untuk mengimplementasikan secara konstruktif prinsip dari teori statistical learning. Dalam kerangka statistical learning, learning artinya mengestimasi sebuah fungsi dari data training. Untuk melakukan ini, sebuah machine learning harus memilih satu fungsi untuk meminimalkan sejumlah risiko yang fungsi estimasinya berbeda dengan fungsi kenyataan (yang belum diketahui). Risiko tergantung pada kompleksitas fungsi-fungsi yang dipilih dari training set. Kemudian learning machine harus menemukan fungsi terbaik, ditentukan oleh kompleksitasnya. SVM cukup populer untuk penggunaan klasifikasi karena kelebihannya antara lain dari segi cara kerja, SVM baik untuk klasifikasi, tidak tergantung pada jumlah fitur dan bisa mengatasi masalah dimensi. Dari segi komputasi, SVM dapat melakukan proses training dengan cepat dan ini berguna dalam teknik learning ketika mengadapi masalah ketidaktegasan.

2.6 Algoritma Decision Tree

2.6.1 Definisi

Decision tree adalah sebuah diagram alir yang berbentuk seperti struktur pohon yang mana setiap *internal node* menyatakan pengujian terhadap suatu atribut, setiap cabang menyatakan output dari pegujian tersebut dan *leaf node* menyatakan kelas–kelas atau distribusi kelas. *Node* yang paling atas disebut sebagai *root node* atau *node* akar. Sebuah *root node* akan memiliki beberapa *edge* keluar tetapi tidak memiliki *edge* masuk, *internal node* akan memiliki satu *edge* masuk dan beberapa *edge* keluar, sedangkan *leaf node* hanya akan memiliki satu *edge* masuk tanpa memiliki *edge* keluar.

Decision tree digunakan untuk mengklasifikasikan suatu sampel data yang belum diketahui kelasnya ke dalam kelas–kelas yang sudah ada. Jalur pengujian data adalah pertama melalui *root node* dan terakhir adalah melalui *leaf node* yang akan menyimpulkan prediksi kelas bagi data tersebut. Atribut data harus berupa data kategorik, bila kontinu maka atribut harus didiskretisasi terlebih dahulu.

2.6.2 Pemilihan Atribut dan Pembentukan tree

Pemilihan atribut untuk menjadi *rootnode* atau *internal node* sebagai *atribut test* berdasarkan atas ukuran *impurity* dari masing–masing atribut. Atribut yang memiliki nilai *impurity* tertinggi akan dipilih sebagai *atribut test.* Ukuran–ukuran *impurity* yang umumnya digunakan adalah i*nformation gain, gain ratio* dan *gini index.*

a. Information Gain

Information Gain merupakan suatu ukuran korelasi pada model parametrik yang menggambarkan ketergantungan antara dua peubah acak X dan Y. Metode splitting optimal point ini biasanya digunakan pada model algoritma ID3 (Gorunescu, 2011).

Information Gain memiliki rumus :

$$\text{Gain(S,A)} = \text{Entropy (S)} - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * \text{Entropy}(S_i) \quad (2.1)$$

$$\text{Entropy(S)} = \sum_{i=1}^{n} - p_i \, log_2(p_i) \quad (2.2)$$

Dimana :

S : himpunan kasus

A : atribut

$|S_i|$: jumlah kasus pada partisi ke-i

$|S|$: jumlah kasus dalam S

n : jumlah partisi atribut

pi : proporsi S_i terhadap S

b. Gain Ratio

Gain Ratio merupakan modifikasi dari information gain untuk mengurangi bias atribut yang memiliki banyak cabang biasanya digunakan pada model algoritma C4.5 (Gorunescu, 2011). Gain ratio memiliki sifat :

1) Bernilai besar bila data menyebar rata
2) Bernilai kecil bila semua data masuk dalam satu cabang

Gain ratio memiliki rumus :

Gain Ratio (S,A) = $\frac{Gain\ (S,A)}{SplitInfo(S,A)}$ (2.3)

Split info (S,A) = - $\sum_{i=1}^{n} \frac{S_i}{S} log_2 \frac{S_i}{S}$ (2.4)

Dimana :

S_i : jumlah sample untuk atribut i

S : ruang (data) sample yang digunakan untuk training

Sedangkan untuk rumus Gain seperti rumus 2.1. Jenis *split* yang dipilih adalah *split* yang memiliki nilai *Gain Ratio* yang terbesar.

c. Gini index

Salah satu kriteria yang dapat digunakan untuk menentukan titik pemecah terbaik (optimal splitting point) adalah GINI index, yang biasanya digunakan dalam algoritma CART (C&RT) dan SPRINT (Gorunescu, 2011:165). Gini index merupakan suatu ukuran ketidaksamaan pada distribusi pendapatan dan memiliki nilai antara 0 sampai 1. Semakin rendah nilai Gini index maka semakin besar pula ukuran kesamaannya. Gini index atributt untuk data dengan m kelas didefinisikan sebagai berikut :

Gini(t) = 1 - $\sum_{i=1}^{n}(p_i)^2$ (2.5)

Dimana :

n : jumlah dari masing-masing atribut

p_i : jumlah atribut dari masing-masing kelas atau labelnya

Bila data dipecah terhadap A menjadi 2 subset D1 dan D2, maka Gini index didefinisikan sebagai berikut :

$Gini_A(D)$ = $\frac{|D1|}{|D|}$Gini(D1) + $\frac{|D2|}{|D|}$Gini(D2) (2.6)

Jenis split terbaik yang dipilih adalah split yang memiliki Gini index terkecil. Ukuran–ukuran tersebut biasanya hanya digunakan pada algoritma tertentu, jadi penentuan ukuran untuk digunakan dalam memilih atribut test sangat di pengaruhi algoritma yang di pilih. Berdasarkan banyaknya edge keluar dari suatu atribut, maka terdapat dua jenis pemisahan yaitu binary split yang menghasilkan dua buah edge keluar dan multyway split yang menghasilkan lebih dari dua edge keluar.

Dalam penggunaannya, kriteria penentuan dengan gini index sering digunakan karena :

- Lebih cocok untuk jumlah partisi atau atribut yang lebih besar.
- Menggunakan proporsi yang di kuadratkan.
- Bisa mengklasifikasikan dengan sempurna, dalam arti lain hasil dari Gini Index akan menjadi nol.
- Terdistribusi merata seperti penentuan nilai Entropy.

2.6.3 Algoritma - algoritma dalam Decision Tree

Ada banyak algoritma pada klasifikasi decision tree ini. Suatu algoritma biasanya dikembangkan untuk meningkatkan kinerja algoritma yang sudah ada. Penentuan algoritma yang terbaik dalam decision tree tentunya tidak bisa ditentukan secara mutlak tetapi sangat tergantung dengan karakteristik training set-nya. Beberapa algoritma decision tree yang cukup populer antara lain : CART, ID3, C4.5, dan C5.0

2.6.4 Algoritma C4.5

Algoritma ini dikembangkan untuk memperbaiki algoritma ID3. Algoritma ini berbasiskan keputusan biner seperti yang terlihat dalam CLS. Jadi selain memiliki karakteristik seperti ID3, C4.5 juga memiliki beberapa karakteristik yang berbeda yang merupakan perbaikan dari karakteristik ID3.

Berikut ini beberapa karakteristik C4.5 yang juga merupakan perbaikan terhadap ID3 :

a. Dapat menangani atribut numerik
b. Dapat menangani *missing value*
c. Melakukan p*runing* untuk memperoleh model yang paling efisien
d. Menggunakan kriteria *gain ratio* untuk menentukan jenis *split* yang terbaik.

2.6.5 Karakteristik Decision Tree

Berikut ini adalah beberapa karakteristik *decision tree* secara umum :

a. *Decision tree* merupakan suatu pendekatan nonparametrik untuk membangun model klasifikasi
b. Teknik yang dikembangkan dalam membangun *decision tree* memungkinkan untuk membangun model secara cepat dari *training set* yang berukuran besar.
c. *Decision tree* dengan ukuran *tree* yang kecil relatif mudah untuk mengi nterpretasinya.

2.7 Algoritma Naïve Bayes

2.7.1 Definisi

Naïve Bayes adalah salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk machine learning dan data mining. Performa naïve bayes yang kompetitif dalam proses klasifikasi walaupun menggunakan asumsi keidependenan atribut (tidak ada kaitan antar atribut). Asumsi keidependenan atribut ini pada data sebenarnya jarang terjadi, namun walaupun asumsi keidependenan atribut tersebut dilanggar performa pengklasifikasian naïve bayes cukup tinggi, hal ini dibuktikan pada berbagai penelitian empiris.

2.7.2 Teorema Bayes

Naïve Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes. Teorema tersebut dikombinasikan dengan ”naive” dimana diasumsikan kondisi antar atribut saling bebas. Berdasarkan teorema tersebut maka perhitungan nilai dari probabilitasnya adalah

$$\text{Posterior} = \frac{\text{Likelihood} \,.\text{Prior}}{\text{Evidence}}$$

Untuk Persamaan dari teorema bayes :

$$P(H|X) = \frac{p(X|H).p\,(H)}{p\,(X)} \quad (2.7)$$

Dimana :

X : data dengan kelas yang belum diketahui

H : Hipotesis data X merupakan kelas spesifik

P(H|X) : Probabilitas hipotesis H berdasar kondisi X (posterior probability)

P(H) : Probilitas hipotesis H (prior probability)

P(X|H) : Probabilitas X berdasar kondisi pada hipotesis H

P(X) : Probabilitas X

Menghitung probabilitas prior untuk tiap kelas (P(H))dengan rumus :

$$P(H) = \frac{Nj}{N} \quad (2.8)$$

Dimana :

Nj : jumlah data pada suatu class

N : jumlah total data

2.7.3 Model Algoritma Naïve Bayes

Naive Bayes Classifier merupakan sekumpulan metode supervised learning yang dapat dikelompokkan berdasarkan distribusi dari menjadi beberapa model/variasi algoritma antara lain :

a. Gaussian Naïve Bayes

Gaussian Naïve Bayes adalah algoritme klasifikasi untuk merepresentasikan probabilitas bersyarat dengan menggunakan nilai yang kontinyu (John & Langley, 2013). Berikut ini akan ditunjukkan rumus Gaussian Naïve Bayes pada Persamaan 2.8

$$P = (X_i = x_i \mid Y_i = y_i) = \frac{1}{\sqrt{2\pi\sigma ij}} e^{\frac{(x_i - u_{ij})^2}{2\,\sigma^2 ij}} \quad (2.9)$$

Dimana :

P : peluang

X_i : atribut ke-i

x_i : nilai atribut ke-i

Y : kelas yang dicari

yj : sub kelas Y yang dicari

u : rata-rata seluruh atribut

σ : deviasi standar dari seluruh atribut

b. Bernoulli Naïve Bayes

Algoritma Bernoulli Naive Bayes mengimplementasikan klasifikasi untuk data yang didistribusikan sesuai dengan distribusi Bernoulli multivariat; yaitu, mungkin terdapat beberapa fitur tetapi masing-masing dianggap sebagai variabel bernilai biner yaitu 0 dan 1. Oleh karena itu, kelas ini membutuhkan sampel untuk direpresentasikan sebagai vektor fitur bernilai biner. Aturan keputusan untuk algoritma Bernoulli Naive Bayes diberikan pada persamaan 2.10 berikut:

$$P(x_i \mid y) = P(i|y)\, x_i + (1 - P(i|y))\,(1 - x_i) \quad (2.10)$$

Dimana :

P (i|y) : probabilitas yang terdapat dalam dokumen kelas C

(1- P (i|y)) : probabilitas yang tidak terdapat dalam dokumen kelas

P (i|y) di hitung dengan persamaan 2.11 yaitu

$$P(i|y) = \frac{df_{i,y}+1}{df_y+2} \quad (2.11)$$

Dimana :

$df_{i,y}$: jumlah dokumen dalam dataset training yang mengandung fitur i dan termasuk dalam kelas y

df_y : jumlah dokumen dataset training yang termasuk dalam kelas C

$+1\ dan\ +2$: parameter laplace smoothing

c. Multinomial Naive Bayes

Multinomial sendiri mengimplementasikan algoritma naive Bayes untuk data yang terdistribusi secara multinomial, dan merupakan salah satu dari dua varian naive Bayes klasik yang digunakan dalam klasifikasi teks di mana data biasanya diwakili sebagai jumlah vektor kata. Distribusi dibatasi oleh vektor x untuk setiap kelas y, di mana jumlah fitur dalam klasifikasi teks dan ukuran kosa kata merupakan probabilitas P(x|y) dari fitur i yang muncul dalam sampel milik kelas y. Penghitungan frekuensi relatif pada multinomial yaitu :

$$P\ (x_i \mid y_i\ ) = \frac{f(x_i \mid y_i\ ) + 1}{|W| + N} \qquad (2.12)$$

Dimana :

$f(x_i \mid y_i\ )$: jumlah kemunculan kata x_i dalam kategori y_i

$|W|$: jumlah kata unik pada semua data latih

N : jumlah semua kata dalam kategori y_i

2.8 Evaluasi dan Validasi Metode Klasifikasi Data Mining

Untuk menguji model, pada penelitian ini, digunakan metode Cross Validation, Confusion Matrix, dan kurva ROC (Receiver Operating Characteristic).

2.8.1 Cross Validation

Cross validation adalah pengujian standar yang dilakukan untuk memprediksi error rate. Data training dibagi secara random ke dalam beberapa bagian dengan perbandingan yang sama kemudian error rate dihitung bagian demi bagian, selanjutnya hitung rata-rata seluruh error rate untuk mendapatkan error rate secara keseluruhan.

2.8.2 Confusion Matrix

Metode ini menggunakan tabel matriks seperti pada Tabel dibawah ini, jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif (Bramer, 2007).

Tabel 2.3 Model Confusion Matrix

Klasifikasi yang benar	**Diklasifikasikan sebagai**	
	+	**-**
+	True Positives (TP)	False Positives (FP)
-	False Negatives (FN)	True Negatives (TN)

True positives adalah jumlah record positif yang diklasifikasikan sebagai positif, false positives adalah jumlah record negatif yang diklasifikasikan sebagai positif, false negatives adalah jumlah record positif yang diklasifikasikan sebagai negatif, true negatives adalah jumlah record negatif yang diklasifikasikan sebagai negative, kemudian masukkan data uji. Setelah data uji dimasukkan ke dalam confusion matrix, hitung nilainilai yang telah dimasukkan tersebut untuk dihitung jumlah sensitivity (recall), specificity, precision dan accuracy. Sensitivity digunakan untuk membandingkan jumlah TP terhadap jumlah record yang positif sedangkan specificity adalah perbandingan jumlah TN terhadap jumlah record yang negatif. Untuk menghitung digunakan persamaan di bawah ini:

$$\text{Precision} = \frac{TP}{TP+FP} * 100\% \quad (2.13)$$

$$\text{Recall} = \frac{TP}{TP+FN} * 100\% \quad (2.14)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (2.15)$$

Dimana :

TP : Jumlah True Positives

FP : Jumlah False Positives

TN : Jumlah True Negatives

FN : Jumlah False Negatives

2.8.3 Kurva ROC

Kurva ROC menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan confusion matrix. ROC adalah grafik dua dimensi dengan false positives sebagai garis horisontal dan true positives sebagai garis vertikal. The area under curve (AUC) dihitung untuk mengukur perbedaan performansi metode yang digunakan. AUC dihitung menggunakan rumus:

$$\text{AUC} = \frac{1}{2}\sum_{i=1}^{n}(x_{i+1} - x_i)(y_{i+1} - y_i) \quad (2.16)$$

Dengan kriteria keakuratan tes diagnostik menggunakan AUC disajikan pada Tabel 2.4 (Gorunescu 2011).

Tabel 2.4 Akurasi AUC

No	Nilai	Kriteria
1.	0.90 – 1.00	Baik Sekali
2.	0.80 – 0.90	Baik
3.	0.70 – 0.80	Cukup
4.	0.60 – 0.70	Kurang
5.	0.50 – 0.60	Gagal

2.9 Normalisasi

Normalisasi merupakan proses penskalaan nilai atribut dari suatu data tertentu menjadi nilai dalam rentang tertentu. Normalisasi dilakukan bertujuan untuk mengurangi adanya kesalahan pada proses data mining. Pada umumnya teknik normalisasi ini dapat dibagi kedalam 5 jenis yaitu : 1) Min-Max, 2) Z-Score, 3) Decimal Scaling, 4) Sigmoidal, 5) Softmax. Salah satu metode yang akan kita bahas yaitu Z-Score, Untuk mencari Z Score atau Nilai Baku ini, kita perlu mengetahui nilai rata-rata (mean) dan stardar deviasi suatu populasi karena Rumus untuk menghitung Z Score adalah dengan mengurangi nilai yang diamati (skor mentah) dengan rata-rata populasi dan kemudia dibagi dengan standar deviasinya. Rumus menghitung Z-Score adalah sebagai berikut :

$$x' = \frac{x - \mu_A}{\sigma_A} \quad (2.17)$$

dimana :

μ : mean

σ : standard deviasi

2.10 Matriks Korelasi

Koefisien korelasi digunakan di dalam statistik untuk mengukur dan menginvestigasi seluruh hubungan antar variable numeric dalam dataset. Fungsi korelasi menggunakan koefisien korelasi Pearson, yang menghasilkan angka antara -1 hingga 1. Hubungan negatif yang kuat ditunjukkan oleh koefisien yang mendekati angka -1 dan korelasi positif yang kuat ditunjukkan oleh koefisien yang mendekati angka 1. Matrik korelasi ini digunakan untuk mereduksi atribut/variabel yang tidak kontributif di dalam dataset.

$$r_{xi.y} = \frac{n(\sum Xi\,Y)-(\sum Xi)\,(\sum Y)}{\sqrt{\{n(\sum x_i^2)-(\sum xi)^2\}\{n(\sum Y^2)-(\sum Y)^2\}}} \quad (2.18)$$

Dimana :

$\sum X_i$: Jumlah data Xi
$\sum Y$: Jumlah dari Y
$\sum X_i Y$: Jumlah dari Xi.Y
$\sum X_i^2$: Jumlah dari X_i^2

Dari hasil yang diperoleh dengan rumus di atas, dapat diketahui tingkat pengaruh variabel X dan variabel Y (Sugiyono, 2013). Pada hakikatnya nilai r dapat bervariasi dari 102 -1 hingga +1, atau secara matematis dapat ditulis menjadi -1 ≤ r ≤ +1. Hasil dari perhitungan akan memberikan tiga alternatif, yaitu: 1. Bila r = 0 atau mendekati 0, maka korelasi antar kedua variabel sangat lemah atau tidak terdapat hubungan antara variabel X terhadap variabel Y. 2. Bila r = +1 atau mendekati +1, maka korelasi antar kedua variabel adalah kuat dan searah, dikatakan positif. 3. Bila r = -1 atau mendekati -1, maka korelasi antar kedua variabel adalah kuat dan berlawanan arah, dikatakan negatif.

2.11 Diskritisasi Variabel

Diskritisasi merupakan salah satu proses transformasi data yang berupa atribut kontinyu menjadi atribut kategoris. Transformasi atribut kontinyu ke atribut kategoris terdiri atas dua langkah, yaitu sebagai berikut:

a. Memutuskan berapa jumlah kategori yang harus digunakan.

b. Memetakan nilai atribut kontinyu ke atribut kategoris.

Pada langkah pertama, setelah diurutkan, nilai atribut kontinyu kemudian dibagi menjadi M interval dengan titik pemotongan M-1.

Pada langkah kedua, semua nilai yang berada dalam setiap interval dipetakan ke nilai kategoris sesuai dengan intervalnya (Prasetyo, 2012: 35)

Ada beberapa algoritma klasifikasi dan clustering yang tidak hanya berurusan dengan atribut nominal dan tidak bisa menangani atribut yang diukur pada skala numerik.

Oleh karena itu, secara umum dataset yang mempunyai atribut numerik harus didiskritisasi ke sejumlah rentang kecil yang berbeda. Meskipun sebagian besar pohon keputusan dan aturan pembelajaran keputusan dapat menangani atribut numerik, namun beberapa implementasi bekerja jauh lebih lambat ketika terdapat atribut numerik. Oleh karena itu, diterapkannya discretization pada atribut numerik yang bersifat continuous (Witten et al., 2011: 314).

2.12 Rapidminer

RapidMiner merupakan perangkat lunak yang bersifat terbuka (open source). RapidMiner adalah sebuah solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi. RapidMiner menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. RapidMiner memiliki kurang lebih 500 operator data mining, termasuk operator untuk input, output, data preprocessing dan visualisasi. RapidMiner merupakan software yang berdiri sendiri untuk analisis data dan sebagai mesin data mining yang dapat diintegrasikan pada produknya sendiri. RapidMiner ditulis dengan munggunakan bahasa java sehingga dapat bekerja di semua sistem operasi. RapidMiner

sebelumnya bernama YALE (Yet Another Learning Environment), dimana versi awalnya mulai dikembangkan pada tahun 2001 oleh RalfKlinkenberg, Ingo Mierswa, dan Simon Fischer di Artificial Intelligence Unit dari University of Dortmund. RapidMiner didistribusikan di bawah lisensi AGPL (GNU Affero General Public License) versi 3. Hingga saat ini telah ribuan aplikasi yang dikembangkan mengunakan RapidMiner di lebih dari 40 negara. RapidMiner sebagai software open source untuk data mining tidak perlu diragukan lagi karena software ini sudah terkemuka di dunia. RapidMiner menempati peringkat pertama sebagai Software data mining pada polling oleh KDnuggets, sebuah portal data-mining pada 2010-2011. RapidMiner menyediakan GUI (Graphic User Interface) untuk merancang sebuah pipeline analitis. GUI ini akan menghasilkan file XML)Extensible Markup Language) yang mendefenisikan proses analitis keingginan pengguna untuk diterpkan ke data. File ini kemudian dibaca oleh RapidMiner untuk menjalankan analis secara otomatis.

a. Sifat

RapidMiner memiliki beberapa sifat sebagai berikut:

1) Ditulis dengan bahasa pemrograman Java sehingga dapat dijalankan di berbagai sistem operasi.
2) Proses penemuan pengetahuan dimodelkan sebagai operator trees
3) Representasi XML internal untuk memastikan format standar pertukaran data.
4) Bahasa scripting memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen.
5) Konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penanganan data.
6) Memiliki GUI, command line mode, dan Java API yang dapat dipanggil dari program lain.

b. Fitur

Beberapa Fitur dari RapidMiner, antara lain:

1) Banyaknya algoritma data mining, seperti decision treee dan self-organization map.
2) Bentuk grafis yang canggih, seperti tumpang tindih diagram histogram, tree chart dan 3D Scatter plots.
3) Banyaknya variasi plugin, seperti text plugin untuk melakukan analisis teks.
4) Menyediakan prosedur data mining dan machine learning termasuk: ETL (extraction, transformation, loading), data preprocessing, visualisasi, modelling dan evaluasi
5) Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI
6) Mengintegrasikan proyek data mining Weka dan statistika R

c. T-test

T-test adalah salah satu operator yang terdapat pada rapidminer yang digunakan untuk menguji perbedaan signifikan pada AUC antar model. T-Test digunakan untuk mengetahui signifikansi perbedaan antar algoritma, dimana kedua algoritma yang akan dikomparasi disatukan dengan operator multiply dan operator t-test untuk menghasilkan nilai. Pada dasarnya metode t-test ini adalah metode pengujian hipotesis dengan menggunakan satu individu (objek penelitian) dengan menggunakan dua perlakuan yang berbeda. Walaupun dengan menggunakan objek yang sama tetapi sampel tetap terbagi menjadi dua yaitu data dengan perlakuan pertama dan data dengan perlakuan kedua.